REMARKS

This Response and Amendment is submitted in complete response to the Office Action mailed December 19, 2002 (hereinafter, the "Office Action"). The Specification has been amended to more clearly define the present invention; no new matter has been added. Claims 1-19 are currently pending in this application. Claims 1, 2, 6, 17 and 19 have been amended to better define the instant invention.

It is believed that no additional claims fees are due or owing in regard to the submission of this Response and Amendment and its attached and related papers. However, if such fees are deemed due, the Office is invited to contact the undersigned at the address and telephone number listed below. A Petition for a three (3) month extension of time together with appropriate fees $930.00 has been being submitted contemporaneously herewith, a grant of the Petition is earnestly requested. Entry of this paper and all amendments and Remarks contained herein is earnestly solicited. Favorable action in response to this paper is earnestly requested, as in a grant of a U.S. Patent.

The paragraphs that follow in conjunction with the amendments to the application presented above are submitted in complete response to this points raised in the Office Action that require such action.

The present invention is directed to the universal problems of upgrading software on a user's PC or other similar computing platforms. The problem with prior art approaches include the interaction required by the user to the PC to obtain and download the software. The present invention solves these problems by providing a unique system for downloading portions of a remote located network object.

<u>Claim Rejections - 35 USC §102(e)</u>

A. Claims 1, 2, 4, 13, 14, 15, 16 and 19 - Bodin et al.:

Claims 1, 2, 4, 13, 14, 15, 16 and 19 are rejected under 35 U.S.C. 102(e) as being anticipated by Bodin et al. **This rejection is respectfully traversed.**

Regarding independent claim 1, the Examiner states, *inter alia*, Bodin et al teaches a system for downloading portions of a remotely located network object, comprising:

a. A server facility configured to be accessed via an electronic data network and to send data corresponding to at least one portion of a network object to a client via the electronic data network (column 2, lines 14-16; column 4, lines 3-11) wherein the server can provide requested files to the user in full or in portions; and

b. A software delegate residing on the client and configured to control an amount of the data and a size of the at least one portion of the network object to be downloaded from the server facility to the client based upon an operating state of the client (column 3, lines 6-10; column 4, lines 1-11) wherein the client must be in operation for the download to take place and must have a connection to the network.

Regarding independent claim 13, the Examiner states, *inter alia*, Bodin et al teaches a system for facilitating downloading portions of a remotely located network object with means for:

a. Using a client computer to access a server facility via an electronic data network (column 2, lines 14-16; column 4, lines 3-11);

b. Receiving, at the client computer, portions of a network object from the server facility (column 4, lines 3-11) wherein the server sends the portions of the file specified by the client;

c. Storing the portions of a network object within the client computer to create a completely downloaded copy of the network object (column 3, lines 44-46) wherein the portion is downloaded to a temporary file; and

d. Controlling a size of the portions of a network object received from the server facility (column 3, lines 6-10; column 4, lines 1-1) wherein the client specifies the size of the file portions to be downloaded.

Initially, it is noted that independent claims 1 and 13 have been amended to more particularly point out and distinctly claim the present invention. In particular, the claims now recite that the network object is downloaded from the server facility independent of the user of the client and based upon an operating state of the client. The new claim language simply makes express what is already implicit in the claims as originally filed and therefore does not and is not intended to limit or otherwise restrict the present invention.

Bodin et al discloses a method and apparatus for passing a large file from a server machine to a client machine as a collection of smaller files. This is achieved without breaking up the large file on the server machine and linking the resultant portions to HTML specific to the downloadable portions. Server code is provided implementing a user interface. The user interface allows a user on the client machine to input the name of the file to be downloaded. A pull-down menu allows the user to specify portion sizes for the file to be downloaded based on the capabilities and transfer rates of the client's machine. Server code dynamically generates a page based on the user input showing the portion sizes which is then displayed to the user at the client machine. Server code next generates download links (Buttons) required to download the file including a download link (Button) to a platform specific shell script or batch file which is dynamically created for the download scenario. The specific shell script or batch file is downloaded to the client machine and the user is allowed to download portions of the large file. After all portions are downloaded, the portions are combined into a single file on the client machine using the shell script, batch file, or other executable .

The Examiner specifically relies on the following teachings of Bodin et al. In Fig. 2 there is shown a display screen 20 where a user selects the size of the portions 26 to download for the file name selected in Fig. 1. In the preferred embodiment, the portions sizes are said to be fixed in value from 1 million bytes (Megs) to 10 million bytes for a file totaling 21,809,820 bytes (col. 3, ls. 5-10). Portion 1 (32) contains the name of a temporary file (e.g., "wEnty100.1") into which those respective portions will be downloaded on the client machine (col. 3, ls. 44-47). According to Bodin et al., one skilled in the art recognizes that the server side processing consists of code which is sensitive to arguments passed from the client machine. More specifically, server code is sensitive to the following arguments:

1) Portion Size
2) Portion Number

The client machine need only send the above arguments to the server machine containing a large downloadable file. The server machine is able to download any large file by performing a seek into the file where the resultant starting point is computed by taking the user selected portion number, subtracting 1, then multiplying by the portion size (e.g., PortionSize*(PortionNumber-1)). At this point, the server machine returns to the client machine the specific portion of the file in the desired PortionSize (col. 3, l. 63 - col. 4, l. 11).

As is readily apparent from the above, Bodin et al does not solve the problems of the prior art in the same or similar manner as the present invention. Specifically, in Bodin et al, the active participation and interaction of the user is still required to download the software. The user must specify portions of the file to be downloaded based on the user's assessment of the capabilities and transfer rates of the client machine. Thus, the download is not independent of the user and based solely on the operating state of the client.

Claim Rejections - 35 USC §103(a)

B. Claims 3, 7, 8, 9 and 10 - Bodin et al. and Young:

Claims 3, 7, 8, 9 and 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bodin et al in view of Young. This rejection is respectfully traversed.

Regarding independent claim 7, the Examiner states, *inter alia*, Bodin et al teach a system for downloading portions of a remotely located network object, comprising:

a. A server facility configured to be accessed via an electronic data network and to send data corresponding to at least one portion of a network object to a client via the electronic data network (column 2, lines 14-16; column 4, lines 3-11) wherein the server can provide requested files to the user in full or in portions;

b. A software delegate capable of residing on the client and configured to control an amount of the data and a size of the at least one portion of the network object to be downloaded from the server facility to the client based upon an operating state of the client (column 3, lines 6-10; column 4, lines 1-11) wherein the client must be in operation for the download to take place and must have a connection to the network; and

c. A client agent, configured to run on an automatic data processing system, to access a storage facility of the automatic data processing system, to access the server facility via the electronic data network to receive data from the server facility via the electronic data network in accordance with the software delegate (column 3, lines 6-10; column 4, lines 1-11) wherein the client can accept data from the server according to the portions specified.

Although the Examiner acknowledges that the system disclosed by Bodin et al fails to disclose means wherein the software delegate is received from the server facility, the Examiner contends that these features are well known in the art and would have been an obvious modification of the system disclosed by Bodin et al as evidenced by Young.

Young is cited by the Examiner as disclosing a system for downloading of files or portions of file involving a software delegate wherein the software delegate is received from the server facility (column 2, lines 48-54) and an applet is used. Applets are downloaded from the server facility.

Given the teaching of Young, the Examiner contends a person having ordinary skill in the art would have readily recognized the desirability and advantages of modifying Bodin et al by employing the download of the software delegate from the server facility. According to the Examiner, this would benefit the system by allowing any client with the proper software to run the application and to access files in portions, and the benefits can more easily be distributed to a wide range of users if applets are used.

Independent claim 7 has been amended in similar manner to independent claims 1 and 13. Thus, independent claim 7 defines over Bodin et al for the same reasons as discussed above with respect to independent claims 1 and 13 and not merely because Bodin et al fails to disclose means wherein the software delegate is received from the server facility.

Young discloses optimizing downloading of files based on which of multiple locations is most efficient in providing the files. In one implementation, portions of a file are downloaded from different servers, and performance data such as a bit rate for each site is used to then select the optimal server to complete the download.

In a further embodiment, an applet intercepts the request for the file from the browser and determines the best server to provide the file. When the request is intercepted, the applet reads a list of available file transfer protocol (ftp) locations from which to download the file. The applet or other type of program then may ping each site to prioritize the list based on shortest response time. A first portion of the file is downloaded from one site, and throughput measurements tracked. Using a reconnect internet protocol command to identify where the first download ended, a second portion of the file is downloaded from the next site on the list, again with throughput measurements tracked. This process is repeated for further portions of the file, and the location with the best throughput is selected to complete the file transfer. In still further embodiments, a same portion such as the first portion is read from each server to ensure Consistent throughput measurements.

In yet a further embodiment, the throughput of the finally selected server is tracked as the file is downloaded. If the throughput drops below a desired throughput, the next best server is selected from the previous list, or the selection and tracking process is started again to determine if a faster server has become available due to changes in demand on the servers.

The Examiner particularly relies on Young's teaching of a standard personal computer system and a module, such as an applet, that intercepts requests from a browser running on the computer system and determines which server from a list of multiple servers is best to provide the file. The applet may also be a module or modules within the browser itself (col. 2, ls. 48-54).

Young does not address the deficiencies of Bodin et al, namely, the implementation of a system for permitting the downloading of software without the active intervention of the PC user and based solely on the operating state of the client. Young pings multiple servers to determine the server(s) best to provide the file. Thus, the present invention is neither disclosed by nor obvious over Bodin et al in view of Young.

Claim Rejections - 35 USC §103(a)

A. Claims 5, 6, 17 and 18 - Bodin et al. and Perlman:

Claims 5, 6, 17 and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bodin et al. This rejection is respectfully traversed.

The Examiner acknowledges that the system disclosed by Bodin et al (as applied to claim 1) fails to disclose means wherein the operating state is an idle state or a busy state. However, the Examiner contends these features are well known in the art and would have been an obvious modification of the system disclosed by Bodin et al as evidenced by Perlman.

Perlman is cited as disclosing a system for download of data over the Internet based on operating state wherein the operating state is an idle state or a busy state (figure 5) wherein data is downloaded when the client is idle and data download is suspended when the client is in use.

Given the teaching of Perlman, the Examiner concludes a person having ordinary skill in the art would have readily recognized the desirability and advantages of modifying Bodin et al by employing the use of the operating system idle time to download data and by employing the stoppage of download when the client is busy. According to the Examiner, this benefits the system by providing the user with the best possible performance while active and for a reliable download during downtime.

Perlman discloses is a method and apparatus for downloading auxiliary data to a client during idle periods and for displaying the auxiliary data while the client is fetching information from the network. According to one embodiment of the present invention, the state of a client device is first determined, wherein the client device is in a fetching state while processing a user request and the user is waiting or in an idle state while not processing a user request and the user is not waiting for the client system. Auxiliary data is then downloaded from a server to the client device when the step of determining determines that the client device is in an idle state. Additionally, the downloaded auxiliary data is buffered in an auxiliary buffer. The auxiliary data is then processed to generate an output and the output is displayed on the client device while the client device is in a fetching state. Fig. 5 is a flow chart illustrating one embodiment of this system.

Accordingly, Pearlman simply discloses a mechanism for taking advantage of the time period when the PC user is a "captive audience" to display advertisement and the like downloaded while the PC is idle. This teaching does not overcome the deficiencies discussed above that exist in Bodin et al's disclosure vis-à-vis the present invention.

Claims 11 and 12 are rejected under 35 U.S.C. 103(a) as being unpatenable over Bodin et al and Young as applied to claim 7 above, and further in view of Perlman. This rejection is respectfully traversed.

Claims 11 and 12 also relate to the situation where the operating state is the idle state or busy state. Thus, the Examiner is directed to the discussion of the rejection of claims 5, 6, 17 and 18 above.

<u>Conclusion</u>:

In view of the foregoing, early and favorable action is respectfully requested, as is the grant of a U.S. Patent.

It is believed that no additional claims fees are due or owing in regard to the submission of this Response and Amendment and its attached and related papers. However, if such fees are deemed due, the Office is invited to contact the undersigned at the address and telephone number listed below. A Petition for a three (3) month extension of time together with appropriate fees $930.00 has been submitted herewith, a grant of the Petition is earnestly requested..

If any additional fees are deemed due or necessary, the Commissioner is hereby authorized to charge any fees due in connection with the present Amendment to Deposit Account 19-4293.

It is believed that a telephonic or in-person interview will in any way expedite examination proceedings related to this application, the Examiner is invited to contact the undersigned attorney of record.

Respectfully submitted,

Erik B. Cherdak
Reg. No. 39,936

STEPTOE & JOHNSON LLP
1330 Connecticut Ave., NW
Washington, DC 20036
(202) 429-6748

Date 6/19/03

APPENDIX

IN THE SPECIFICATION:

Page 4, amend the second full paragraph at lines 13-25 as follows:

--According to another aspect of the present invention, [providing] provided is a system for downloading portions of a remotely located network object. The system includes a server facility, a software delegate and a client. The server facility is configured to store and to serve the software delegate and a network object. The server facility is further configured to be accessed via an electronic data network, such as the Internet and WWW. The client is configured to access the server facility via the electronic data network and to download and execute the software delegate. The client is further configured to download portions of the network object in accordance with the executed software delegate, until a completely downloaded copy of the network object is stored locally on the client.--

IN THE CLAIMS:

2. (currently amended) A system for downloading portions of a remotely located network object, comprising:

a client;

a server facility configured to be accessed via an electronic data network by said client and to send data corresponding to at least one portion of a network object to [a] said client via said electronic data network; and a software delegate residing on said client and configured to control an amount of said data and a size of said at least one portion of said network object to be downloaded from said server facility to said client independent of a user of said client and based solely upon an operating state of said client.

2. (original) The system according to claim 1, wherein said electronic data network is the Internet.

3. (original) The system according to claim 1, wherein said software delegate is a Javascript applet.

4. (original) The system according to claim 1, wherein said amount of data is a range of bytes and said size of said at least one portion is dependent on said operating state.

5. (original) The system according to claim 1, wherein said operating state is an idle state.

6. (original) The system according to claim 1, wherein said operating state is a busy state.

7. (currently amended) A system for downloading portions of a remotely located network object, comprising:

a client;

a server facility configured to be accessed via an electronic data network by said client and to send data corresponding to at least one portion of a network object to a client via said electronic data network; [and]

a software delegate capable of residing on said client and configured to control an amount of said data and a size of said at least one portion of said network object to be downloaded from said server facility to said client independent of a user of said client and based solely upon an operating state of said client; and

a client agent, configured to run on an automatic data processing system, to access a storage facility of said automatic data processing system, to access said server facility via said electronic data network to receive said software delegate, and to receive data from said server facility via said electronic data network in accordance with said software delegate.

8. (original) The system according to claim 7, wherein said client agent is an Internet browser and said electronic data network is the Internet.

9. (original) The system according to according to claim 7, wherein said software delegate is a Javascript applet.

10. (original) The system according to claim 7, wherein said amount of data is a range of bytes and said size of at least one portion is dependent on said operating state.

11. (original) The system according to claim 7, wherein said operating state is an idle state.

12. (original) The system according to claim 7, wherein said operating state is a busy state.

13. (currently amended) A method for facilitating downloading portions of a remotely located network object comprising the steps of:

using a client computer to access a server facility via an electronic data network;

receiving, at said client computer, portions of a network object from said server facility;

storing said portions of [a] said network object within said client computer to create locally a completely downloaded copy of said network object [is created locally]; and

controlling a size of said portions of [a] said network object received from said server facility independent of a user of said client computer and based solely upon an operating state of said client computer.

14. (original) The method according to claim 13, wherein said electronic data network is the Internet.

15. (currently amended) The method according to claim 13, wherein said size of said portions of [a] said network object in said controlling step is a range of bytes.

16. (currently amended) The method according to claim 13, wherein said size of said portions of [a] said network object in said controlling step is dependent on [an] said operating state of said client computer.

17. (original) The method according to claim 16, wherein said operating state is an idle state.

18. (original) The method according to claim 16, wherein said operating state is a busy state.

19. (currently amended) The method according to claim 13, wherein said controlling step is performed by a software delegate residing on said client computer.